

A New Formal Context for Symmetric Dependencies

Jaume Baixeries

Departament de Llenguatges i Sistemes Informàtics.
Universitat Politècnica de Catalunya.
08024 Barcelona. Catalonia.
jbaixer@lsi.upc.edu

Abstract. In this paper we present a new formal context for symmetric dependencies. We study its properties and compare it with previous approaches. We also discuss how this new context may open the door to solve some open problems for symmetric dependencies.

1 Introduction and Motivation

In database theory there are different types of dependencies, yet, two of them appear to be the most popular: functional dependencies and multivalued dependencies. The reason is that both dependencies come handy in order to explain the normalization of a database scheme. But some of these dependencies are not only confined to the database domain. For instance, implications (the equivalent of functional dependencies for binary data) are present in datamining and learning ([4,19,20]).

In general terms, a dependency states a relationship between sets of attributes in a table. Let us suppose that we have the following set of attributes: $\mathcal{U} = \{name, income, age\}$ in a table that contains the following records:

id	Name	Income	Age
1	Smith	30.000	26-10-1956
2	Hart	35.000	14-02-1966
3	Smith	30.000	02-01-1964

In such a case, we have that the relationship between *age* and the attributes *income* and *name* is functional, this is, that given a value of *age*, the value of *income* and *name* can be determined. We also have that the value of *name* can be determined by *income* and viceversa. In such a case, given these functional relationships between the attributes, we say that the functional dependencies $age \rightarrow \{name, income\}$, $name \rightarrow income$ and $income \rightarrow name$ hold in that table.

Functional dependencies and multivalued dependencies have their own set of axioms ([9,21]), which state what dependencies hold in the presence of other dependencies. For instance, an axiom for functional dependencies states that

transitivity holds, which means that, in the previous case, if we had that $name \rightarrow income$ and $income \rightarrow age$ hold in that table (which is not true in that table, but just as a supposition), it must follow necessarily that $name \rightarrow age$ holds. Given a set of dependencies Σ , we define as Σ^+ the set of all dependencies that hold according to those axioms.

These axioms, in turn, are also shared by other dependencies: implications share the same axioms of functional dependencies ([4]), and degenerate multi-valued dependencies share the same axioms of multivalued dependencies ([5]). That is why we generically call Armstrong Dependencies (AD) those dependencies that share the same axioms of the former, and Symmetric Dependencies (SD) those that share the axioms of the latter.

Since in this paper we are focusing on the syntactical properties of those dependencies, we will only talk of Armstrong and symmetric dependencies, rather than functional or multivalued dependencies.

The lattice characterization of a set of Armstrong dependencies has been widely studied in [10,11,13,14,15], and their characterization with a formal context in [7,17]. However, the lattice characterization of symmetric dependencies has not been so widely studied. The main work is in [12], and the characterization of symmetric dependencies with a formal contexts was studied in [3,5] (we talk indistinctly of a lattice characterization and a characterization with a formal context). In the case of AD's, the formal context yields a powerset lattice ([17]), whereas in the case of symmetric dependencies, it yields a partition lattice ([3]).

The fact that some problems related to AD's have been solved using their lattice characterization, suggests that the same problems for SD's could also be solved using their corresponding lattice characterization. We name three of those problems already solved for AD's, not yet for SD's: **learning** SD's, the **finding of a minimal basis** for a set of dependencies for SD's and the **characterization of mixed sets** of SD's and AD's.

In general terms, **query learning** consists in deducing a function (a formula) via membership queries to an oracle. This method has been used to learn sets of Horn clauses, which can also be seen as implications ([8]), or, more generally, sets of Armstrong dependencies. Thus, the same general algorithm for learning Horn clauses ([1]) has been adapted to learn Armstrong dependencies ([2]). This adaptation was obviously eased by the fact that Horn clauses and Armstrong dependencies share the same set of axioms. Yet, no such algorithm for symmetric dependencies exists (to the best of the author's knowledge).

The **minimal base** (also: Duquenne-Guigues basis [16]) is the minimal set of Armstrong dependencies needed to compute Σ^+ . In [11], [16] and [17] it is characterized and computed in terms of the (powerset) lattice characterization of Σ^+ .

We have been dealing with unmixed sets of AD's and SD's, but there exists an **axiomatizations of mixed sets of AD's and SD's** ([21]), but no lattice characterization of mixed sets.

Although AD's and SD's are related, the lattice characterization yielded by the formal context in [3] is quite different in nature to that for AD's. *Potentially*, it may pose different problems. The first is that the solutions that have been found for AD's (based on their lattice characterization) may not be applied directly to the case of SD's. We do not mean that having AD's characterized with a powerset lattice and SD's with a partition lattice makes it impossible to solve the same problems for SD's. What we mean is that having a similar characterization for SD's would make it easier to try and find an answer using existing solutions for AD's.

A second drawback is that the size of the formal context for SD's is much larger, in comparison with that for AD's. This may cause a problem in case the context is used in practical applications, but, more importantly, there are partitions that play no rôle in that characterization. A simple analysis of [3] yields that partitions that contain no singleton are completely useless, but a more detailed analysis (out of the scope of this paper) indicates that there are more redundant partitions.

Finally, although partitions may be intuitive when dealing with SD's, *they do not reflect the $B \Leftrightarrow \neg B$ symmetry of the definition of symmetric dependencies* (as stated by Alan Day in [12]). It seems that the connection between AD's and SD's is stronger than what the partition lattice characterization suggests.

As a step towards solving the learning problem and computation of a minimal basis for SD's as well as the characterization of mixed sets of SD's and AD's, in this paper we present a new formal context for symmetric dependencies, following the work started in [3]. The results presented in this paper parallel those results, but from a different perspective that, we think, improve both the understanding and the possibilities to solve the open problems previously listed.

This paper starts with the Notation section, followed by a Previous Work section that explains the departing point of this paper. In the Results section, we present a new formal context for SD's. We also present an example in a separate section to illustrate the results. Finally, we discuss some aspects of this new formal context and present the conclusions and future work.

2 Notation

We depart from a set of attributes \mathcal{U} . We use non capital letters for single elements of that set, starting with a, b, c, \dots , and capital letters for subsets of \mathcal{U} .

The complement of a set $X \subseteq \mathcal{U}$ is \bar{X} . We drop the union operator and use juxtaposition to indicate set union. For instance, instead of $X \cup Y$ we write XY . Generally, we also drop the set notation, and write abc instead of $\{a, b, c\}$.

We define the powerset of a set \mathcal{U} as $\mathcal{P}(\mathcal{U})$. The set of partitions that can be formed with \mathcal{U} is $\text{Part}(\mathcal{U})$. The notation for a partition is $P = [P_1 \mid P_2 \mid \dots \mid P_n]$, where P_i are the classes (subsets) of P . If needed, we indicate that the attributes in a set X are in fact a set of singletons with this notation: \underline{X} . For instance, $\underline{\{a, b, c, d\}} = \{\{a\}, \{b\}, \{c\}, \{d\}\}$. We overload $P \geq Q$ to indicate that a

partition P refines a partition Q and $P \leq Q$ to indicate that P is coarser than Q . More details of this (reversed) order can be found in [18].

As for Formal Concept Analysis, we use the usual notation ([17]), which includes the use of $'$ as the (overloaded) function that relates the set of attributes and that of objects and viceversa.

2.1 Symmetric Dependencies

A symmetric dependency is a relation between two sets of attributes, and it is stated as $X \Rightarrow Y$. Given a set of attributes \mathcal{U} , we define $SD_{\mathcal{U}}$ as the set of all symmetric dependencies that can be formed with \mathcal{U} . Although they will only be mentioned in this paper, we say that $X \rightarrow Y$ is an Armstrong dependency. Given a set of SD's $\Sigma \subseteq SD_{\mathcal{U}}$, we say that the **closure** of Σ is Σ^+ , and consists of Σ plus the set of all SD's that can be derived from Σ applying recursively the following axioms:

Definition 1 (Axioms for SD's).

1. *Reflexivity:* If $Y \subseteq X$, then, $X \Rightarrow Y$ holds.
2. *Complementation:* If $X \Rightarrow Y$ holds, then, $X \Rightarrow \overline{XY}$ holds.
3. *Augmentation:* If $X \Rightarrow Y$ holds and $W' \subseteq W \subseteq \mathcal{U}$, then, $XW \Rightarrow YW'$ holds.
4. *Transitivity:* If $X \Rightarrow Y$ and $Y \Rightarrow Z$ hold, then, $X \Rightarrow Z \setminus Y$ holds.

Because of complementation, we give a symmetric dependency as $X \Rightarrow Y \mid Z$, where $Z = \overline{XY}$. We always assume that the rightest set in the right-hand side of a symmetric dependency is the complementary of the union of the other two. However, sometimes we will state it explicitly, as in $X \Rightarrow Y \mid \overline{XY}$ and sometimes we will simply use $X \Rightarrow Y \mid Z$. In both cases, X is the left-hand side of the dependency, Y its first right-hand side, and Z its second right-hand side. The set $SD_{\mathcal{U}}$ is the set of all non-trivial symmetric dependencies that can be formed using all the attributes in \mathcal{U} . By non-trivial we mean those SD's $X \Rightarrow Y \mid Z$ such that:

Definition 2. A symmetric dependency $X \Rightarrow Y \mid Z$ is non-trivial if:

1. $X \cup Y \cup Z = \mathcal{U}$.
2. $X \cap Y = X \cap Z = Y \cap Z = \emptyset$.
3. $X \neq \emptyset, Y \neq \emptyset, Z \neq \emptyset$.

As it can be seen, according to the axioms for symmetric dependencies, this limitation incurs in no loss of information, since the remaining symmetric dependencies can easily be derived from $SD_{\mathcal{U}}$ ([21]).

It is precisely the complementation rule that states the relation between Armstrong dependencies and symmetric dependencies. Broadly speaking, we could say that a symmetric dependency $X \Rightarrow Y \mid Z$ is equivalent to the fact that either the Armstrong dependencies $X \rightarrow Y$ or $X \rightarrow Z$ hold. This is a too general statement, but if, as an example, we take, functional dependencies and its symmetric

counterpart, degenerate multivalued dependencies, we see that the definition of a functional dependency $X \rightarrow Y$ states that whenever two tuples agree on X they also agree on Y , whereas the definition of a degenerate multivalued dependency $X \rightrightarrows Y \mid Z$ states that whenever two tuples agree on X they also agree on Y or they agree in Z . In fact, there are also a set of two axioms in the case we are dealing with mixed sets of AD's and SD's. One of this axioms state that if $X \rightarrow Y$ holds, then, $X \rightrightarrows Y \mid \overline{XY}$ holds as well. This example is just to indicate that the relationship between AD's and SD's is strong, and that SD's can be as a generalization of AD's.

Given a set of symmetric dependencies Σ , we say that the **dependency basis** of a set of attributes $X \subseteq \mathcal{U}$ (that is: $DB_{\Sigma}(X)$) is the coarsest partition of \mathcal{U} such that all the dependencies $X \rightrightarrows Y \mid Z$ that hold in Σ^+ are those such that Y (symmetrically Z) is the union of one or more classes of $DB(X)$. This partition always exists ([21]) and defines all the symmetric dependencies that hold in Σ^+ such that their left-hand side is X .

We also have that, since reflexivity holds for SD's, all the attributes of $X \subseteq \mathcal{U}$ are singletons in $DB_{\Sigma}(X)$.

2.2 Previous Work

The origins of defining a formal context to characterize the closure of a set of Armstrong dependencies started in [17]. This formal context was defined as:

$$\mathbb{K}_{AD}(\mathcal{U}) = (AD_{\mathcal{U}}, \wp(\mathcal{U}), I)$$

where $AD_{\mathcal{U}}$ is the set of Armstrong dependencies that can be formed with the set of attributes \mathcal{U} , and I was a binary relation between an Armstrong dependency and a set of attributes.

In [3], it was presented a formal context for symmetric dependencies with identical properties:

$$\mathbb{K}_{SD}(\mathcal{U}) = (SD_{\mathcal{U}}, \text{Part}(\mathcal{U}), I')$$

The relations I and I' are generically called "respect" relations: a set of attributes (a partition) respects an Armstrong (symmetric) dependency.

Both formal contexts, in spite of its obvious structural differences, characterized the closure of a set of dependencies of its kind. In fact, both contexts provided the following results for each respective kind of dependencies:

1. $\Sigma^+ = \Sigma''$.
2. Σ' is the lattice characterization of Σ^+ .

When we say that Σ' was the lattice characterization of Σ^+ , it may seem redundant, since we already have that $\Sigma^+ = \Sigma''$. What we mean is that Σ' alone, without the application of the operator $'$, also characterized all the dependencies of Σ^+ . This was done with the definition of a closure operator on Σ' :

$$\Gamma_{\Sigma'}(X) = \bigwedge \{ Y \in \Sigma' \mid Y \supseteq X \}$$

The fact that this function is total indicates that \wedge is always defined in Σ' . Depending on the formal context we were dealing with, we would have that $X \in \wp(\mathcal{U})$ (AD's) or that $X \in \text{Part}(\mathcal{U})$ (SD's). In the case of Armstrong dependencies, we would then have that $X \rightarrow Y \in \Sigma^+$ if and only if:

$$\Gamma_{\Sigma'}(X) = \Gamma_{\Sigma'}(XY)$$

In the case of a symmetric dependency, it is a little bit more elaborated from a syntactical point of view, yet, equivalent to the previous case: $X \Rightarrow Y \mid Z \in \Sigma^+$ if and only if:

$$\Gamma_{\Sigma'}([\underline{X} \mid YZ]) = \Gamma_{\Sigma'}([\underline{X} \mid Y \mid Z])$$

Clearly, Σ' alone gives us the information of which dependencies are in Σ^+ by querying the (closure) operator $\Gamma_{\Sigma'}$. In both cases, and oversimplifying, we can say that a dependency holds in Σ^+ if and only if there is some kind of relationship between its left-hand side and its right-hand side, being this relationship defined by the formal context.

3 Results

The results in this paper try to overcome the potential problems that may represent the different nature of the current formal contexts for AD's and SD's (powerset versus partitions), as well as the larger size of a partition set, by presenting a characterization of symmetric dependencies based on a formal context whose set of attributes is the powerset of \mathcal{U} instead of its partitions. This context will generalize that for AD's in [17] as it will be seen in Section 5.

We define a formal context, and prove that it characterizes the set of symmetric dependencies Σ^+ , in a way similar to that in [3]: $(SD_{\mathcal{U}}, \wp(\mathcal{U}), \mathbf{I})$, where the relation \mathbf{I} is defined as follows:

Definition 3. $A \subseteq \mathcal{U}$ *respects* a symmetric dependency $X \Rightarrow Y \mid Z$ (that is: $X \Rightarrow Y \mid Z \mathbf{I} A$) if and only if:

$$A \not\supseteq X \text{ or } A \supseteq XY \text{ or } A \supseteq XZ$$

We have that $\Sigma' \subseteq \wp(\mathcal{U})$. As a trivial consequence of Definition 3 we have the following proposition:

Proposition 1. $X \Rightarrow Y \mid Z \in \Sigma''$ if and only if

$$\nexists A \in \Sigma' : A \supseteq X \text{ and } A \not\supseteq XY \text{ and } A \not\supseteq XZ$$

We now study the properties of this contexts and how they characterize Σ^+ . We first see that all the dependencies that are in Σ^+ are also present in Σ'' . To prove this claim, we must prove axiom by axiom that the dependencies derived by those axioms are also present in Σ'' , but since reflexivity and complementation are trivial, we only prove augmentation and transitivity.

Proposition 2 (Augmentation). *If $X \Rightarrow Y \mid \overline{XY} \in \Sigma$, and $W' \subseteq W$ then, $XW \Rightarrow YW' \mid \overline{XWY} \in \Sigma''$.*

Proof. By the way of contradiction, we suppose that there is a set $A \subseteq \mathcal{U}$, $A \in \Sigma'$ such that (note that $XW\overline{XWY} = XW\overline{Y}$)

$$A \supseteq XW \text{ and } A \not\supseteq XWY \text{ and } A \not\supseteq XW\overline{Y}$$

Since $X \Rightarrow Y \mid \overline{XY} \in \Sigma$, we have that $A \not\supseteq X$ or $A \supseteq XY$ or $A \supseteq X\overline{Y}$ (because $X\overline{XY} = X\overline{Y}$). We have that $A \supseteq XW$ discards $A \not\supseteq X$. So, we only have two possible options:

- (i) $A \supseteq XY$, which in combination with $A \supseteq XW$ yields $A \supseteq XYW$, which contradicts $A \not\supseteq XWY$.
- (ii) $A \supseteq X\overline{Y}$, which in combination with $A \supseteq XW$ yields $A \supseteq XW\overline{Y}$, which contradicts $A \not\supseteq XW\overline{Y}$.

■

Proposition 3 (Transitivity). *If $X \Rightarrow Y \mid \overline{XY} \in \Sigma$ and $Y \Rightarrow Z \mid \overline{YZ} \in \Sigma$, then, $X \Rightarrow Z \setminus Y \mid \overline{X(Z \setminus Y)} \in \Sigma$.*

Proof. By the way of contradiction, we suppose that there is $A \subseteq \mathcal{U}$, $A \in \Sigma'$ such that

$$A \supseteq X \text{ and } A \not\supseteq X(Z \setminus Y) \text{ and } A \not\supseteq X\overline{X(Z \setminus Y)}$$

We have to note that $X\overline{X(Z \setminus Y)} = X\overline{(Z \setminus Y)}$, and that $\overline{Z \setminus Y} = Y\overline{Z}$, we finally have that $X\overline{X(Z \setminus Y)} = XY\overline{Z}$. Therefore, we suppose that there is a set $A \subseteq \mathcal{U}$, $A \in \Sigma'$ such that:

- (i) $A \supseteq X$.
- (ii) $A \not\supseteq X(Z \setminus Y)$.
- (iii) $A \not\supseteq XY\overline{Z}$.

On the other hand, we have that:

$X \Rightarrow Y \mid \overline{XY} \in \Sigma$ implies that $A \not\supseteq X$ or $A \supseteq XY$ or $A \supseteq X\overline{Y}$.

$Y \Rightarrow Z \mid \overline{YZ} \in \Sigma$ implies that $A \not\supseteq Y$ or $A \supseteq YZ$ or $A \supseteq Y\overline{Z}$.

Since we are assuming that $A \supseteq X$, we can discard $A \not\supseteq X$. We also have that the case $A \supseteq XY$ discards $A \not\supseteq Y$. This leaves three possibilities, either:

- (i) $A \supseteq XY$ and $A \supseteq YZ$, that is, $A \supseteq XYZ \supseteq X(Z \setminus Y)$. This contradicts $A \not\supseteq X(Z \setminus Y)$.
- (ii) $A \supseteq XY$ and $A \supseteq Y\bar{Z}$, that is, $A \supseteq XY\bar{Z}$. This contradicts $A \not\supseteq XY\bar{Z}$.
- (iii) $A \supseteq X\bar{Y}$. $Y \cap Z = \emptyset$ implies that $Z \setminus Y \subseteq \bar{Y}$. All this yields $A \supseteq X\bar{Y} \supseteq X(Z \setminus Y)$. This contradicts $A \not\supseteq X(Z \setminus Y)$.

■

Therefore, we have proved that any Σ'' contains, at least, all the symmetric dependencies that are in Σ^+ .

Corollary 1. $\Sigma^+ \subseteq \Sigma''$.

Proof. By Propositions 2 and 3. ■

We now prove completeness, that is, that Σ'' **only** contains all the dependencies in Σ^+ .

Theorem 1. $\Sigma'' \subseteq \Sigma^+$.

Proof. We prove that $X \Rightarrow Y \mid Z \notin \Sigma^+$ implies that $X \Rightarrow Y \mid Z \notin \Sigma''$.

We have that $X \Rightarrow Y \mid Z \notin \Sigma^+$. It means that the dependency basis of X is such that in $DB_\Sigma(X) = [\underline{X} \mid P_1 \mid \dots \mid P_n]$ (with $n \geq 1$) there is, at least, a class P_k such that $P_k \cap Y \neq \emptyset$ and $P_k \cap Z \neq \emptyset$. We fix P_k in this proof. We note that $|P_k| \geq 2$, since it contains, at least, one attribute from Y and one from Z .

Let $P = (\bigcup_{j=1}^n P_j) \setminus P_k$, that is, P is the union of all partitions in $DB_\Sigma(X)$ which are not X , except P_k . Therefore, $XP = \overline{P_k}$. We now claim that $XP \in \Sigma'$. We prove this statement by the way of contradiction. Assume that $XP \notin \Sigma'$. That is because there is a dependency $R \Rightarrow S \mid T \in \Sigma$ such that $X \supseteq R$ and $X \not\supseteq RS$ and $X \not\supseteq RT$. This implies that there is, at least, one attribute in RS which is not in XP , and, at least, one attribute in RT which is not in XP . Let them be $s \in RS, s \notin XP$ and $t \in RT, t \notin XP$. Since $X \supseteq R$, then, $s \in S, s \notin XP$ and $t \in T, t \notin XP$. Necessarily, since $s, t \notin XP$, then, $s, t \in P_k$.

Since $XP \supseteq R$, by reflexivity, $XP \Rightarrow R \mid \overline{XP}R$, and by transitivity, $XP \Rightarrow S \setminus R \mid \overline{XP}(S \setminus R)$. Without lack of generality, we assume that R, S, T are disjoint, so, finally, we have $XP \Rightarrow S \mid \overline{XP}S$. By the definition of $DB_\Sigma(X)$, then, $X \Rightarrow P \mid \overline{XP}$, and by transitivity, we have $X \Rightarrow S \setminus XP \mid \overline{X}(S \setminus XP)$. Since $s \in S, s \notin XP$, then, $s \in S \setminus XP$, and since $t \in T$ (assuming R, S, T disjoint), $t \notin S$, that, together with $t \notin XP$ yields that $t \in \overline{X}(S \setminus XP)$. It means that the attributes s, t are in different classes in $DB_\Sigma(X)$, but this contradicts the previous assumption that $P_k \in DB_\Sigma(X)$.

Now, we have that $XP \in \Sigma'$. Since $s, t \notin XP$, we have that $XP \not\supseteq XY$ and $XP \not\supseteq XZ$ and $XP \supseteq X$, which implies $X \Rightarrow Y \mid Z \notin \Sigma''$.

■

We have that Σ'' is exactly the set Σ^+ . But, as we have already discussed in the previous section, in [3] and [17] we had a method to query Σ' whether a dependency was in Σ^+ , and consisted in the closure operator $\Gamma_{\Sigma'}$ that, given a set of attributes, returned the meet of its up-set. In this present case, we may have that Σ' is not a lattice (but a partial lattice) and the same operator would not be a total function. Therefore, we use the up-set, instead of its meet:

Definition 4. *Let $\Sigma \subseteq SD_{\mathcal{U}}$. We define the up-set of $X \subseteq \mathcal{U}$ as follows:*

$$UP_{\Sigma}(X) = \{Y \in \Sigma' \mid Y \supseteq X\}$$

This definition is the standard one in lattice theory ([18]) when Σ' is an ordered set. The proof of the following proposition is trivial, yet, it will come handy to prove the last result of this paper.

Proposition 4. *Let $X, Y, Z \subseteq \mathcal{U}$ such that $Y \supseteq X$ and $Z \supseteq X$.*

$$UP_{\Sigma}(X) = UP_{\Sigma}(Y) \cup UP_{\Sigma}(Z)$$

if and only if

$$\nexists A \in \Sigma' : A \supseteq X \text{ and } A \not\supseteq Y \text{ and } A \not\supseteq Z$$

We need to remark that, although the set Σ' may not be closed under set intersection, the set of all up-sets of Σ' is closed under intersection. We are now ready to prove that it can be tested whether a dependency is in Σ^+ querying Σ' alone:

Proposition 5. *$X \Rightarrow Y \mid Z \in \Sigma^+$ if and only if*

$$UP_{\Sigma}(X) = UP_{\Sigma}(XY) \cup UP_{\Sigma}(XZ)$$

Proof.

$$X \Rightarrow Y \mid Z \in \Sigma^+$$

if and only if (by Corollary 1 and Theorem 1)

$$X \Rightarrow Y \mid Z \in \Sigma''$$

if and only if (by Proposition 1)

$$\nexists A : A \supseteq X \text{ and } A \not\supseteq XY \text{ and } A \not\supseteq XZ$$

if and only if (by Proposition 4)

$$UP_{\Sigma}(X) = UP_{\Sigma}(XY) \cup UP_{\Sigma}(XZ)$$

■

4 Example

We provide a running example in order to illustrate and clarify the results that are contained in the previous section. We depart from a set of attributes $\mathcal{U} = \{a, b, c, d\}$. The resulting formal context is presented in Figure 1.

As stated in Theorem 1, this context computes the set Σ^+ . For instance, let us take the set

$$\Sigma = \{a \Rightarrow b \mid cd, b \Rightarrow ad \mid c\}$$

According to this context, we have that

$$\Sigma' = \{c, d, bc, cd, abc, abd, acd, bcd, abcd\}$$

and, finally,

$$\begin{aligned} \Sigma'' = \Sigma^+ = \{ & a \Rightarrow b \mid cd, b \Rightarrow ad \mid c, a \Rightarrow c \mid bd, a \Rightarrow d \mid bc, \\ & ab \Rightarrow c \mid d, ac \Rightarrow b \mid d, ad \Rightarrow b \mid c, bd \Rightarrow a \mid c \} \end{aligned}$$

To check these results, we see that $ac \Rightarrow b \mid d$, $ad \Rightarrow b \mid c$ and $bd \Rightarrow a \mid c$ are derived from Σ by the reflexivity, transitivity and complementation. For instance, given $a \Rightarrow b \mid cd$, by reflexivity we have $ac \Rightarrow a \mid bd$, and by transitivity $ac \Rightarrow b \mid d$ (complementation comes from the notation $X \Rightarrow Y \mid Z$ used in this paper). Dependencies $ad \Rightarrow b \mid c$ and $bd \Rightarrow a \mid c$ can be derived alike.

As for the remaining SD's:

$$a \Rightarrow c \mid bd, a \Rightarrow d \mid bc$$

by applying transitivity to $a \Rightarrow b \mid cd$ and $b \Rightarrow ad \mid c$, we obtain $a \Rightarrow c \mid bd$, and with complementation we have $a \Rightarrow d \mid bc$.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>ab</i>	<i>ac</i>	<i>ad</i>	<i>bc</i>	<i>bd</i>	<i>cd</i>	<i>abc</i>	<i>abd</i>	<i>acd</i>	<i>bcd</i>	<i>abcd</i>
<i>a</i> ⇒ <i>b</i> <i>cd</i>	×	×	×	×	×			×	×	×	×	×	×	×	×
<i>b</i> ⇒ <i>a</i> <i>cd</i>	×		×	×	×	×	×			×	×	×	×	×	×
<i>a</i> ⇒ <i>c</i> <i>bd</i>	×	×	×	×	×			×	×	×	×	×	×	×	×
<i>c</i> ⇒ <i>a</i> <i>bd</i>	×	×		×	×	×	×			×	×	×	×	×	×
<i>a</i> ⇒ <i>d</i> <i>cb</i>	×	×	×				×	×	×	×	×	×	×	×	×
<i>d</i> ⇒ <i>a</i> <i>cb</i>	×	×	×		×	×	×			×	×	×	×	×	×
<i>b</i> ⇒ <i>c</i> <i>ad</i>	×		×	×		×	×	×		×	×	×	×	×	×
<i>c</i> ⇒ <i>b</i> <i>ad</i>	×	×		×	×		×	×	×		×	×	×	×	×
<i>b</i> ⇒ <i>d</i> <i>ac</i>	×		×	×		×	×			×	×	×	×	×	×
<i>d</i> ⇒ <i>b</i> <i>ac</i>	×	×	×		×	×		×	×		×	×	×	×	×
<i>c</i> ⇒ <i>d</i> <i>ab</i>	×	×		×	×		×		×	×	×	×	×	×	×
<i>d</i> ⇒ <i>c</i> <i>ab</i>	×	×	×		×	×		×		×	×	×	×	×	×
<i>ab</i> ⇒ <i>c</i> <i>d</i>	×	×	×	×		×	×	×	×	×	×	×	×	×	×
<i>ac</i> ⇒ <i>b</i> <i>d</i>	×	×	×	×	×		×	×	×	×	×	×	×	×	×
<i>bc</i> ⇒ <i>a</i> <i>d</i>	×	×	×	×	×	×		×	×	×	×	×	×	×	×
<i>ad</i> ⇒ <i>b</i> <i>c</i>	×	×	×	×	×	×		×	×	×	×	×	×	×	×
<i>bd</i> ⇒ <i>a</i> <i>c</i>	×	×	×	×	×	×	×		×	×	×	×	×	×	×
<i>cd</i> ⇒ <i>a</i> <i>b</i>	×	×	×	×	×	×	×	×		×	×	×	×	×	×

Fig. 1. Formal context $(SD_{\mathcal{U}}, \wp(\mathcal{U}), I)$ for $\mathcal{U} = \{a, b, c, d\}$

We now present an example with one more attribute, which may provide more insight in the details, but in this case, we do not present the context explicitly. The set of attributes is now $\mathcal{U} = \{a, b, c, d, e\}$. Let Σ be the set of symmetric dependencies:

$$\begin{aligned}
 & b \Rightarrow a \mid cde \quad b \Rightarrow c \mid ade \quad c \Rightarrow a \mid bde \quad c \Rightarrow b \mid ade \quad d \Rightarrow a \mid bce \quad d \Rightarrow e \mid abc \\
 & e \Rightarrow a \mid bcd \quad e \Rightarrow d \mid abc
 \end{aligned}$$

According the formal context $(SD_{\mathcal{U}}, \wp(\mathcal{U}), I)$, we have that:

$$\Sigma' = \{a, abc, ade, abcd, abce, abde, acde, bcde, abcde\}$$

We can see that, applying the axioms of symmetric dependencies in Definition 1, the set Σ^+ is:

$$\begin{aligned}
 & b \Rightarrow a \mid cde \quad b \Rightarrow c \mid ade \quad c \Rightarrow a \mid bde \quad c \Rightarrow b \mid ade \quad d \Rightarrow a \mid bce \quad d \Rightarrow e \mid abc \\
 & e \Rightarrow a \mid bcd \quad e \Rightarrow d \mid abc \quad abd \Rightarrow c \mid e \quad acd \Rightarrow b \mid e \quad ce \Rightarrow a \mid bd \quad de \Rightarrow a \mid bc \\
 & bcd \Rightarrow a \mid e \quad abe \Rightarrow c \mid d \quad ace \Rightarrow b \mid d \quad bce \Rightarrow a \mid d \quad bde \Rightarrow a \mid c \quad cde \Rightarrow a \mid b \\
 & ab \Rightarrow c \mid de \quad ac \Rightarrow b \mid de \quad ad \Rightarrow bc \mid e \quad ae \Rightarrow bc \mid d \quad bc \Rightarrow a \mid de \quad bd \Rightarrow ac \mid e \\
 & bd \Rightarrow a \mid ce \quad bd \Rightarrow ae \mid c \quad be \Rightarrow ac \mid d \quad be \Rightarrow ad \mid c \quad be \Rightarrow a \mid cd \quad cd \Rightarrow a \mid be \\
 & cd \Rightarrow ae \mid b \quad cd \Rightarrow ab \mid e \quad ce \Rightarrow ab \mid d \quad ce \Rightarrow ad \mid b
 \end{aligned}$$

We only state the non-trivial dependencies as in Definition 2. We take, for instance, the dependencies:

$$bd \Rightarrow ac \mid e, bd \Rightarrow a \mid ce, bd \Rightarrow ae \mid c$$

They are derived from the dependencies $b \Rightarrow a \mid cde$ and $b \Rightarrow c \mid ade$. They are in Σ^+ because the sets that include bd are $abcd, abde, bcde, abcde$. This obviously means that all of them respect all the dependencies in Σ^+ . We take, for instance, the set $abcd$ and see that it respects $bd \Rightarrow ae \mid c$ because $abcd \geq bcd$ (the left-hand side plus the second right-hand side) and that it also respects $bd \Rightarrow a \mid ce$ because $abcd \geq abd$ (the left-hand side plus the first right-hand side). We can see in this example the duality of the definition of the relation respect. This is one case of derivation by augmentation, which means that the dependencies that derive another dependency remove the sets that would prevent the derived dependency from appearing in Σ'' . In this latter particular case, the sets that could be forbidding any of these dependencies from appearing in Σ'' have been cleared by $b \Rightarrow a \mid cde$ and $b \Rightarrow c \mid ade$. We take, for instance, the set bde , (which would prevent $bd \Rightarrow a \mid ce$ from being in Σ'') is not in Σ' because it does not respect the dependency $b \Rightarrow a \mid cde$.

We now illustrate one case of derivation by transitivity with the following set:

$$a \Rightarrow bc \mid de, bc \Rightarrow d \mid ae$$

By transitivity, we have that $a \Rightarrow d \mid bce \in \Sigma^+$. If we take $\Sigma = \{a \Rightarrow bc \mid de\}$, we have:

$$\Sigma' = \{b, c, d, e, bc, bd, be, cd, ce, de, abc, ade, bcd, bce, bde, cde, abcd, abce, abde, acde, bcde, abcde\}$$

It is clear that $a \Rightarrow d \mid bce \notin \Sigma''$ since the sets $abc, acde \in \Sigma'$ do not respect this dependency. Now, if we include $bc \Rightarrow d \mid ae$ in Σ , we have:

$$\Sigma' = \{b, c, d, e, bd, be, cd, ce, de, ade, bcd, bde, cde, abcd, abce, abde, acde, bcde, abcde\}$$

It has precisely been the dependency $bc \Rightarrow d \mid ae$ the one that has cleared both abc and $acde$ from Σ' and, therefore, allows $a \Rightarrow d \mid bce$ to appear in Σ'' .

We now illustrate how Σ' alone can be used to query what dependencies hold in Σ^+ . Again, we have that Σ is the set:

$$\begin{array}{l} b \Rightarrow a \mid cde \quad b \Rightarrow c \mid ade \quad c \Rightarrow a \mid bde \quad c \Rightarrow b \mid ade \quad d \Rightarrow a \mid bce \quad d \Rightarrow e \mid abc \\ e \Rightarrow a \mid bcd \quad e \Rightarrow d \mid abc \end{array}$$

and, therefore:

$$\Sigma' = \{a, abc, ade, abcd, abce, abde, acde, bcde, abcde\}$$

We can see that Σ' is not closed ($abcd, abde \in \Sigma'$, but $ab \notin \Sigma'$). Now, suppose that we want to test whether a dependency is in Σ^+ . For instance, we take a dependency that is not in Σ^+ , as $a \Rightarrow bc \mid de$ and query Σ' :

$$\begin{aligned} UP_{\Sigma}(a) &= \{ a, abc, ade, abcd, abce, abde, acde, abcde \} \\ UP_{\Sigma}(abc) &= \{ abc, abcd, abce, abcde \} \\ UP_{\Sigma}(ade) &= \{ ade, abde, acde, abcde \} \end{aligned}$$

According to Proposition 5, since the sets $UP_{\Sigma}(a)$ and $UP_{\Sigma}(abc) \cup UP_{\Sigma}(ade)$ do not coincide, then, this dependency does not hold in Σ^+ . We see that the set that does not allow this equality to hold is the set a , which is in Σ' because all dependencies in Σ are respected by this set. We take now a positive example of a dependency that is in Σ^+ but not in Σ , as for instance $ab \Rightarrow c \mid de$:

$$\begin{aligned} UP_{\Sigma}(ab) &= \{ abc, abcd, abce, abde, abcde \} \\ UP_{\Sigma}(abc) &= \{ abc, abcd, abce, abcde \} \\ UP_{\Sigma}(abde) &= \{ abde, abcde \} \end{aligned}$$

In this case, the sets $UP_{\Sigma}(ab)$ and $UP_{\Sigma}(abc) \cup UP_{\Sigma}(abde)$ coincide.

5 Discussion

We have seen in Section 2 that the different characterizations of dependencies with formal contexts follow a common pattern, regardless of the type of dependencies or the definition of the context. Yet, the definition of formal contexts for AD's and SD's as in [3] was structurally different (powersets versus partitions) and that made it difficult to find a relationship and generalization between both contexts, in spite of the clear structural similarities that exist between AD's and SD's.

Now, we have that the relation \mathbf{I} is a **generalization** of the relation defined in the context $\mathbb{K}_{AD}(\mathcal{U}) = (AD_{\mathcal{U}}, \wp(\mathcal{U}), I)$. We recall the definition of this relation ([17]):

Definition 5. $A \subseteq \mathcal{U}$ *respects* an Armstrong dependency $X \rightarrow Y$ iff:

$$A \not\supseteq X \text{ or } A \supseteq XY$$

We see that this definition avoids the reference to the second right-hand side, precisely because in AD's, complementation does not hold. If we drop this part from Definition 3, we have the definition of the respect relation for Armstrong dependencies.

This generalization seems to suggest that the solutions that have been developed based on the lattice characterization of sets of Armstrong dependencies, may also be applied to symmetric dependencies, namely:

1. To define a formal context for mixed sets of AD's and SD's.

2. To adapt the classical query algorithm for learning Armstrong dependencies.
3. To characterize the generating set of a set of symmetric dependencies.

Yet, although we are now in a better position to attack those problems, it does not seem to be a trivial task. For instance, the intuition would tell us that defining a formal context for mixed sets of dependencies, such that the relation would be the union of the relations already defined for AD's and SD's would work, but this is not the case. In fact, although this is out of the scope of this paper, this mixed formal contexts characterizes the symmetric dependencies that are in Σ^+ , where Σ is a mixed set of AD's and SD's, but fails in characterizing the AD's that are in Σ^+ . However, this simple strategy allows to advance towards the definition of a mixed formal context, which would have not been that simple departing from a partition context.

Adapting the classic learning algorithm for learning AD's and the characterization of the generating set of a set of SD's may encounter some difficulties. The main difference between Σ' for Armstrong and symmetric dependencies is that for the former, Σ' is always a powerset lattice closed under intersection, whereas for symmetric dependencies, this is not necessarily the case, and, therefore, not all the existing solutions for Armstrong dependencies, based on lattices, may be applied out of the box to symmetric dependencies. Yet, the fact that now we are dealing with contexts of the same nature, offers a much clearer perspective and understanding than before.

It must be said too that whereas this new characterization may make it potentially easier to find methods for finding minimal basis and query learning for SD's, it is true that SD's have not yet been used outside the database domain. We think that advancing in the study of lattice characterization for SD's and finding algorithmic similarities with FD's may introduce the use of SD's in other domains, namely knowledge discovery and machine learning, or in database theory, where it is already present: it would be of interest to have algorithms to compute minimal basis for SD's, profiting from the important collection of algorithms that compute the minimal basis of a set of AD's.

Finally, we would like to remark that the size of the formal context is greatly improved w.r.t the context in [3], since we have replaced the set $\text{Part}(\mathcal{U})$ by the set $\mathcal{P}(\mathcal{U})$. Yet, and for the sake of algorithmic solutions already existing in the FCA community, we have to say that the size of the context remains exponential.

6 Conclusions and Future Work

We have presented a new formal context for symmetric dependencies. This contexts provides the same functionalities as previous approaches, and it is much simpler. Yet, it offers the same expressivity power and, in fact, reduces the conceptual gap between Armstrong and symmetric dependencies that existed in a previous approach. We strongly believe that this may be the first step towards the resolution via formal concept analysis, of the learning, minimal bases and mixed sets of dependencies problems for symmetric dependencies, profiting from solutions already existing for Armstrong dependencies.

References

1. Angluin D., Frazier M., Pitt L. Learning Conjunctions of Horn Clauses. *Machine Learning*, 9:147-164, 1992.
2. Arias M., Balcázar, José L. Canonical Horn Representations and Query Learning. *Lecture notes in computer science*, vol. 5809, p. 156-17, 2009.
3. Baixeries, Jaume. A Formal Context for Symmetric Dependencies. *ICFCA 2008*. LNAI 4933.
4. Baixeries, Jaume and Balcázar, José L. Discrete Deterministic Data Mining as Knowledge Compilation. *Proceedings of Workshop on Discrete Mathematics and Data Mining in SIAM International Conference on Data Mining*, 2003.
5. Baixeries, Jaume and Balcázar, José L. Characterization and Armstrong Relations for Degenerate Multivalued Dependencies Using Formal Concept Analysis. *Formal Concept Analysis, Third International Conference, ICFCA 2005*, Lens, France, February 14-18, 2005, *Proceedings. Lecture Notes in Computer Science*, 2005
6. Baixeries, Jaume and Balcázar, José L. Unified Characterization of Symmetric Dependencies with Lattices. *Contributions to ICFCA 2006. 4th International Conference on Formal Concept Analysis 2005*.
7. Baixeries, Jaume. A Formal Concept Analysis framework to model functional dependencies. *Mathematical Methods for Learning*, 2004.
8. Balcázar, José L. and Baixeries, Jaume. Discrete Deterministic Data Mining as Knowledge Compilation. *Workshop on Discrete Mathematics and Data Mining in SIAM Int. Conf.* 2003.
9. Beeri, Catriel and Fagin, Roland and Howard, John H. A Complete Axiomatization for Functional and Multivalued Dependencies in Database Relations. *Proceedings of the 1977 ACM SIGMOD International Conference on Management of Data*, Toronto, Canada, August 3-5, 1977.
10. Caspard, Nathalie and Monjardet, Bernard. The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: a Survey. *Proceedings of the 1998 Conference on Ordinal and Symbolic Data Analysis (OSDA-98)*. *Discrete Applied Mathematics*, 2003.
11. Day, Alan. The Lattice Theory of Functional Dependencies and Normal Decompositions. *International Journal of Algebra and Computation* Vol. 2, No. 4 409-431. 1992.
12. Day, Alan. A Lattice Interpretation of Database Dependencies. *Semantics of Programming Languages and Model Theory*, 1993.
13. Demetrovics, János and Hencsey, Gusztav and Libkin, Leonid and Muchnik, Ilya. Normal Form Relation Schemes: a New Characterization. *Acta Cybernetica*, 1992.
14. Demetrovics, János and Huy, Xuan. Representation of Closure for Functional, Multivalued and Join Dependencies. *Computers and Artificial Intelligence*, 1992.
15. Demetrovics, János and Libkin, Leonid and Muchnik, Ilya. Functional Dependencies in Relational Databases: a Lattice Point of View. *Discrete Applied Mathematics*, 1992.
16. Duquenne, Vincent and Guigues, J.L. Familles Minimales d'Implications Informatives Resultant d'un Tableau de Données Binaires. *Mathematics and Social Sciences*, 1986.
17. Ganter, Bernhard and Wille, Rudolf. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
18. Grätzer, George. *General Lattice Theory*. Academic Press, 1978.

19. Pfaltz, John L. Using Concept Lattices to Uncover Causal Dependencies in Software. Formal Concept Analysis, 4th International Conference, ICFCA 2006, Dresden, Germany, February 13-17, 2006.
20. Pfaltz, John L. Incremental Transformation of Lattices: A Key to Effective Knowledge Discovery. In *Proc. of the First Intl. Conf. on Graph Transformation (ICGT'02)*, pages 351-362, Barcelona, Spain, Oct 2002.
21. Ullman, Jeffrey D. Principles of Database Systems. Computer Science Press, 1982.